Tut_5

Aufgabe 1: Nadaraya-Watson "zu Fuß" und mit ksmooth()

```
miete <- read.table("http://chris.userweb.mwn.de/statistikbuch/mietspiegel2015.txt", header =
miete <- miete[miete$wfl <= 200, ]
attach(miete)
```

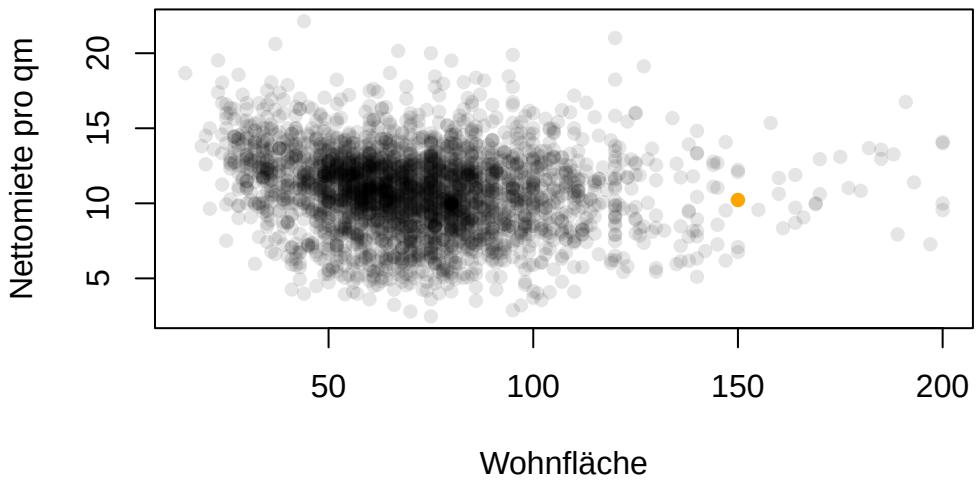
a)

Erstellen Sie ein Streudiagramm zwischen den Variablen Wohnfläche ("wfl") und Nettomiete pro Quadratmeter ("nmqm"). Wir betrachten anschließend das nichtparametrische Regressionsmodell

$$\text{nmqm}_i = m(\text{wfli}_i) + \epsilon_i, \quad i = 1, \dots, n$$

Bestimmen Sie "zu Fuß" — d.h. ohne Zuhilfenahme einer R-Funktion zur nichtparametrischen Regression — den Wert des Nadaraya-Watson-Schätzers ˆ m(wfl) an der Stelle wfl = 150 unter Verwendung des Rechteckkerns und mit Bandweite b = 20. Ergänzen Sie anschließend den berechneten Schätzwert wie folgt im Streudiagramm:

```
plot(wfl, nmqm, pch = 16, col = "#0000001A", xlab = "Wohnfläche",
     ylab = "Nettomiete pro qm")

I <- nmqm[wfl >= 140 & wfl <= 160]

m_hat_150 <- mean(I)

points(150, m_hat_150, col = "orange", pch = 16)
```



b)

Definieren Sie nun mit den folgenden Befehlen eine Funktion in R, welche unserem Rechteckkern entspricht:

```
K <- function(z){ifelse(abs(z)>0.5, 0, 1)}
```

Probieren Sie die Funktion einmal aus, indem Sie sie auf verschiedene Werte von z anwenden. Bestimmen Sie anschließend noch einmal den Wert des Nadaraya-Watson-Schätzers $\hat{m}(wfl)$ an der Stelle Wfl = 150 (mit b =20), diesmal unter Verwendung der Funktion K() (siehe Slide 14 der Vorlesung).

$$\hat{m}(x) = \frac{\sum_{i=1}^{n} K\left(\frac{x-x_i}{b}\right) y_i}{\sum_{i=1}^{n} K\left(\frac{x-x_i}{b}\right)}, \quad \text{wobei} \quad K(y) = \begin{cases} 1 & \text{falls } -\frac{1}{2} \leq y \leq \frac{1}{2}, \\ 0 & \text{sonst.} \end{cases}$$

```
K(5)
```

```
[1] 0
```

```r
K(0.5001)
```

```
[1] 0
```

```r
K(0.5)
```

```
[1] 1
```

```r
K(0.4)
```

```
[1] 1
```

```r
sum(K((150 - wfl) / 20) * nmqm) / sum(K((150 - wfl) / 20))
```

```
[1] 10.22276
```

```r
m_hat_150
```

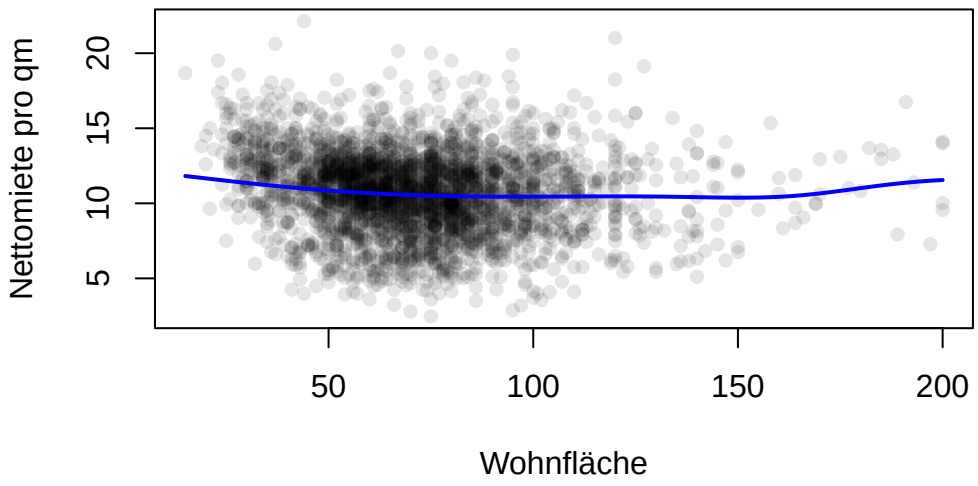
```
[1] 10.22276
```

c)

Sie können den Nadaraya-Watson-Schätzer mit Gaußkern in komplett analoger Art und Weise bestimmen, indem Sie statt der von uns definierten Funktion K() (Rechteckkern) einfach dnorm() (Gaußkern) nutzen. Probieren Sie dies einmal aus, wieder an der Stelle wfl = 150 und wieder mit Bandweite b = 20. Ergänzen Sie anschließend mit: points()

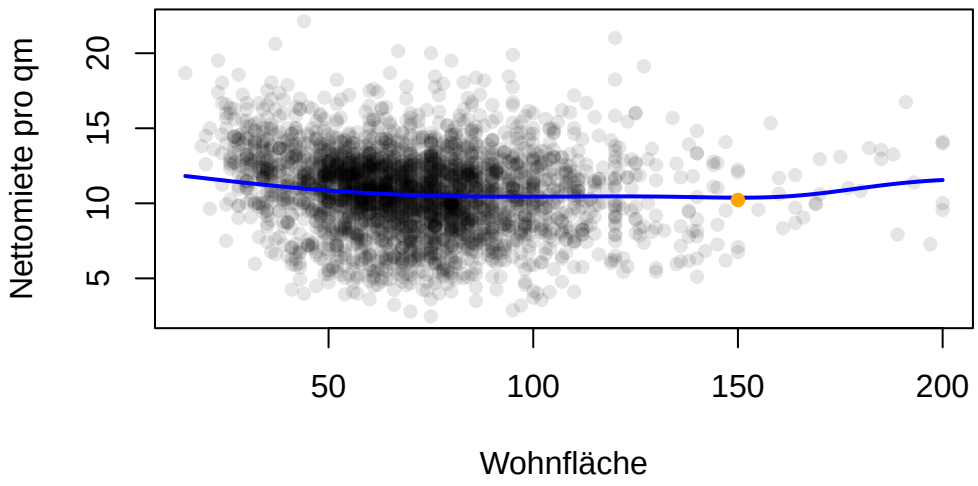
Fügen Sie mit:

```r
plot(wfl, nmqm, pch = 16, col = "#0000001A", xlab = "Wohnfläche",
     ylab = "Nettomiete pro qm")

NWS <- ksmooth(wfl, nmqm, kernel = "normal", bandwidth = 55)

lines(NWS$x, NWS$y, col = "blue", lwd = 2)
```



Hinweis: Die Funktion ksmooth() benutzt eine Umrechnung der Bandweiten, welche hier gerade so ist, dass die Wahl von bandwidth = 55 etwa unserer Wahl b = 20 entspricht.

```
m_hat_norm <- sum(dnorm((150 - wfl) / 20) * nmqm) / sum(dnorm((150 - wfl) / 20))

plot(wfl, nmqm, pch = 16, col = "#0000001A", xlab = "Wohnfläche",
     ylab = "Nettomiete pro qm")

lines(NWS$x, NWS$y, col = "blue", lwd = 2)

points(150, m_hat_150, col = "orange", pch = 16)
```



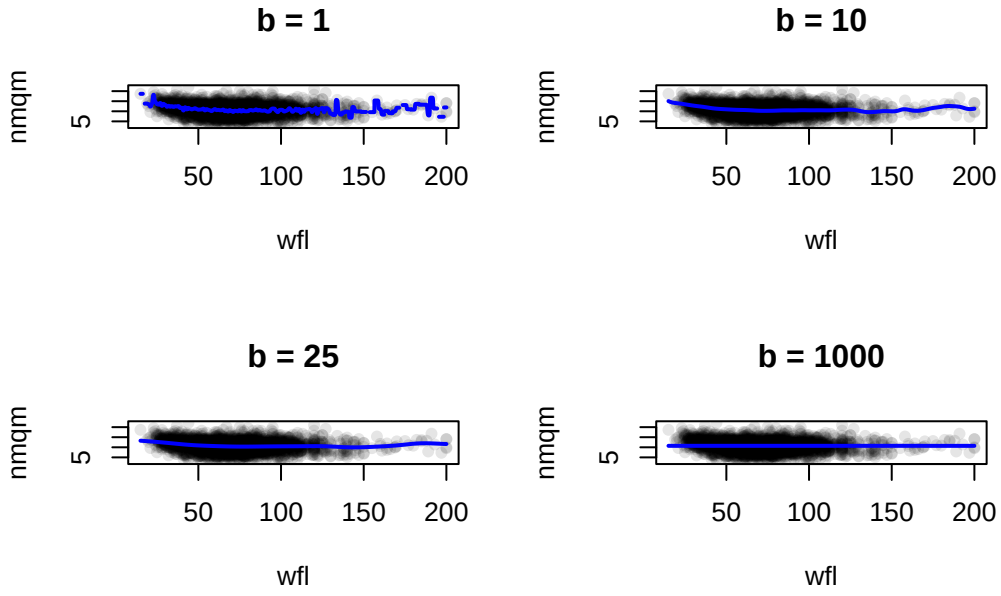
Aufgabe 2: Bandweitenwahl und Kreuzvalidierung

a)

Schauen Sie sich mit ksmooth() den Nadaraya-Watson-Schätzer für das Modell aus Auf gabe 1 mit dem Gaußkern für verschiedene Bandweiten an und wählen Sie eine Ihres Erachtens geeignete Bandweite aus.

```r
par(mfrow = c(2, 2))

# Plot 1
plot(wfl, nmqm, pch = 16, col = "#0000001A", main = "b = 1")
NWS <- ksmooth(wfl, nmqm, kernel = "normal", bandwidth = 1)
lines(NWS$x, NWS$y, col = "blue", lwd = 2)

# Plot 2
plot(wfl, nmqm, pch = 16, col = "#0000001A", main = "b = 10")
NWS <- ksmooth(wfl, nmqm, kernel = "normal", bandwidth = 10)
lines(NWS$x, NWS$y, col = "blue", lwd = 2)

# Plot 3
```

```r
plot(wfl, nmqm, pch = 16, col = "#0000001A", main = "b = 25")
NWS <- ksmooth(wfl, nmqm, kernel = "normal", bandwidth = 25)
lines(NWS$x, NWS$y, col = "blue", lwd = 2)

# Plot 4
plot(wfl, nmqm, pch = 16, col = "#0000001A", main = "b = 1000")
NWS <- ksmooth(wfl, nmqm, kernel = "normal", bandwidth = 1000)
lines(NWS$x, NWS$y, col = "blue", lwd = 2)
```



b)

Wir wollen nun eine Kreuzvalidierung umsetzen. Gehen Sie wie folgt vor:

1.

Bestimmen Sie mit ksmooth() den Ausdruck

$$\left(\text{nmqm}_i - \hat{m}_{b,-i}(\text{wfli}_i)\right)^2$$

aus dem Kreuzvalidierungskriterium für $i = 1$, d.h. wir betrachten zunächst nur die erste Beobachtung zur Validierung. Verwenden Sie den Gaußkern und eine Bandweite von 10. Zur Anpassung müssen Sie die erste Beobachtung unberücksichtigt lassen, z.B. durch:

```
# ksmooth(miete$wfl[-1], miete$nmqm[-1], ...)
```

Durch Ergänzung der Option x.points = miete$wfl[1] wird dann die Vorher sage für eben jene erste Beobachtung bestimmt.

```
cv_i <- ksmooth(wfl[-1], nmqm[-1], kernel = "normal", bandwidth = 10,
 x.points = wfl[1])$y

(nmqm[1] - cv_i)^2
```

```
[1] 2.561681
```

2.

Bestimmen Sie nun

$$CV(10) = \sum_{i=1}^{n} \left(\text{nmqm}_i - \hat{m}_{10,-i}(\text{wfli}_i)\right)^2,$$

d.h. den Wert des Kreuz validierungskriteriums für = 10, nun aber für die Summe aller Beobachtungen — dazu eignet sich eine for-Schleife.

```
n <- nrow(miete)
CV <- rep(NA, n)

for(i in 1:n){
 mcv <- ksmooth(wfl[-i], nmqm[-i],
 kernel = "normal",
 bandwidth = 10,
 x.points = wfl[i])$y
 CV[i] <- (nmqm[i]- mcv)^2
}

sum(CV)
```

```
[1] 20556.2
```

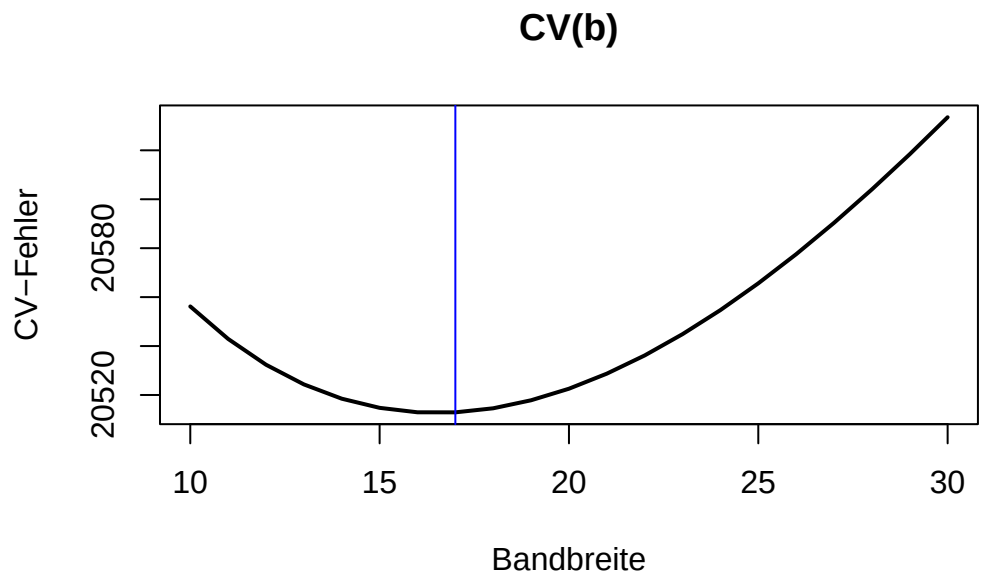
3.

Erweitern Sie ihre for-Schleife aus 2., sodass Sie CV(b) für die Bandweiten 10,10.1,10.2,...,19.9,20 ausrechnen. Plotten Sie dann CV(b) als Funktion von b. Wählen Sie die gemäß des Kreuzvalidierungskriteriums beste Bandweite und vergleichen Sie den Wert mit Ihrer subjektiven Wahl aus a).

```
b <- seq(10, 30, by = 1) # Bei Sequenzen < 1 kann die Berechnung sehr lange dauern
nb <- length(b)
n <- length(nmqm)
CVb <- rep(NA, nb)

for(k in 1:nb){
  CV <- rep(NA, n)
  for (i in 1:n){
    mcv <- ksmooth(wfl[-i], nmqm[-i], kernel = "normal",
                   bandwidth = b[k], x.points = wfl[i])$y
    CV[i] <- (nmqm[i] - mcv)^2
  }
  CVb[k] <- sum(CV)
}

b[which.min(CVb)]
```

```
[1] 17
```

```
plot(b, CVb, type = "l", main = "CV(b)", xlab = "Bandbreite", ylab = "CV-Fehler", lwd = 2)
abline(v = 17, lwd = 1, col = "blue")
```



Das optimale b liegt bei etwa 17.

c)

Inwiefern äußert sich in dem Plot aus 3. der Bias-Varianz-Trade-off?

1. CV ist hoch für kleines b, da dann die Varianz groß ist
2. CV ist hoch für großes b, da dann der Bias groß ist
3. CV ist am niedrigsten für moderates b (der Kompromiss zwischen den Extremen)